

第1章 回路設計者として身を立てるための“土台”を築く

――基礎をしっかり学ぶこと，つねに新しい視点を持ち続けること

外村元伸

長年，半導体メーカにおいてCPUの開発に携わった先輩エンジニアが回路設計者としての心構えを示す．ハードウェア記述言語（HDL）で記述ファイルを作成し，論理合成ツールを適用すれば，ひととおりの回路は作れるものである．しかし，論理回路の基本を押さえていないと，回路の簡素化や性能向上，低消費電力化などの最適化を行うことができない．（編集部）

筆者は，CPU（central processing unit）のLSI（large scale integrated circuit）を設計・製造するメーカに長く在籍し，おもに演算器とその回路設計技術の研究・開発に携わってきました．

● 回路設計技術の教育が時代にマッチしていない

今日，CMOS（complementary metal oxide semiconductor）回路設計はあたりまえになっていますが，従来プロセス[注1]からCMOSプロセスへの移行期にはCMOS回路をターゲットにした演算回路に関する設計技術は確立していませんでした．当時は回路を設計しようにも，初期のコンピュータ（銀行などで使われている汎用大型計算機．「メインフレーム」と呼ばれている）時代に考え出された演算器に関する書籍や論文くらいしか参考となる資料がありませんでした．そのため，みずから演算方式の基礎的なところから検討し，CMOS回路に適した演算回路方式を考案していかなければなりませんでした．

こうした回路設計について現在はどうかというと，やはり設計技術について詳細に解説された良書があまりないように思います．設計現場にいても，関係者がほんとうに演算器やその回路についての知識を備えているのかあやしい，というのが筆者の実感です．教育現場でも本質的な回路設計（とくに演算回路設計）について語られていないのが現状

注1：以前はバイポーラ（低速TTL，高速ECLに分かれる）プロセス，過渡期にはBiCMOS（バイポーラ＋CMOS混載；高速と低消費電力を両立）プロセス，また消費電力を気にしないNMOSのみの時代もあった．



図1
回路設計の教育は現状にマッチしているか？
大学でも開発の現場でも，昔から同じようなテキストを使って回路設計（とくに演算回路）の教育がなされているように思う．時代に合わせて，教育内容は変えていかなくては….

KeyWord　演算器，2進数，HDL，RTL，CMOS，ASIC，FPGA，線形補間，特許，ブース・エンコーダ

Column 回路設計と特許

エンジニアたるもの，どのような技術に対しても特許の存在を無視することはできません．たとえ独自に導き出した技術でも，また容易なものであっても，あなどってはなりません．その技術に対してすでにだれかが特許をとっているかもしれません．難しい複雑な技術よりも，だれもが容易に思いつく技術のほうが，特許として多くの面で効力を発揮します．

筆者も直接的および間接的に特許に絡んだ事件に巻き込まれた経験がいくつかあります．個人とかベンチャ企業が所有する特許についてはクロスライセンス契約を結びにくいので，とくにやっかいです．例えば，線形補間演算器（詳細は第2章を参照）に関連する特許は，**表A-1**に示すように特許出願されて成立しているものもあれば，成立していないものもあります．一般に，米国の特許は成立しやすいのですが，**表A-1**に示すようにずいぶん古い公知例があり，容易に類推できることが見逃されがちです．

エンジニアのみなさんが特許調査を行いたいときには，以下のホームページを気軽に参照できます[(2)]．

- 米国特許商標庁（http://www.uspto.gov/）
- ヨーロッパ特許庁（http://ep.espacenet.com/）
- 日本国特許庁特許電子図書館（http://www.ipdl.ncipi.go.jp/homepg.ipdl）

表A-1 線形補間演算器関連特許のリスト

特許番号	名　称	出願社	特許成立日（あるいは公開日）	備　考
US 3,412,240	Linear Interpolater	Gen Prec Systems社	1968年11月19日	2の補数表現を利用した式あり
US 5,113,362	Integrated Interpolation and Method of Operation	Analog Devices社	1992年5月12日	論文あり．成立から17年有効
特開平6-83852	アルファ合成演算器	松下電器産業	1994年3月25日	拒絶査定
US 5,517,437 特開平7-13741	Alpha Blending Calculator アルファ合成演算器	松下電器産業	1996年5月14日 1995年1月17日	日本は拒絶査定

ではないでしょうか（**図1**）[注2]．

歴史的な経緯から，乗算器には今でもブース・エンコーダという方式が使われることがあります．また，ブース・エンコーダについて解説されている本もよく目にします．この方式は，3ビット列（000，001，010，011，100，101，110，111）の2進数を，2けた（$\bar{1}0$，$0\bar{1}$，00，01，10）の符号付き2進数列に変換して演算を行います（ここで，$\bar{1}=-1$）．3ビット列の場合と比べて非ゼロのけたが2個から1個に減っているため，あたかも演算器の規模が半分になったように思われるかもしれません．しかし，実際には2進数列への変換でもゲートが必要なため，トータルのゲート規模は半減しません．しかも，この方法は理論もテクニックも複雑で，あまりお勧めできません．現在では，カウンタを利用する方式のほうが簡単ですし，拡張性もあります．

● 一歩踏み込んで考える姿勢がたいせつ

人は0～9までの10進数をベースにして計算を行いますが，コンピュータをはじめとするディジタル回路では電気的に実現しやすい‘0’と‘1’の状態を利用して2進数をベースに数を表現し，加減乗除算などの演算を行います．「2進数をベースに」といっても，小学校のときに習った10進数の扱いとあまり変わらない部分がほとんどです．このように割り切って考えれば，じつに単純に扱えるのが演算器です．

しかし，演算器を実現するための回路設計と周辺アルゴリズムに目を向ければ，さらに奥深い世界が開けてきます．例えば，ゲーム機を実現するための演算器設計を考えてみましょう．最近では3次元コンピュータ・グラフィックスのための演算が必要になります．モデルの動きを表現するための3次元座標変換や影を付けるためのシェーディング（描画）演算などがあります．これらの演算処理には，かならず決められた時間内に演算結果を出すことが要求されます．このような要求に対して，シミュレーションの数値計算でよく使われる，いつ演算が完了するか予測できないループを伴う反復解法は使えません．決められた演算精度と時間で結果を出すためには，近似計算が必要になります．近似の対象になる計算式は非線形の場合が多く，確立された方法がないときは設計者が独自に考えなければならないこともあります．

じつは，筆者が演算器設計に真剣に興味を持つようになったのは，今から20年以上も前に，3次元コンピュー

注2：最近，何人かの研究者の長年の努力により，回路設計の方向性が見えてきている．最新技術に詳しい本も出版された．参考文献（1）を参照していただきたい．



図2
現在のRTL設計
演算器の具体的な構成技術について詳しく知らなくても，ツール・ベンダなどが提供する加減乗除算程度の演算器のライブラリを利用すれば，なんとか回路は設計できるのだが….

タ・グラフィックスの試作機を立ち上げたことがきっかけです（世界的にもかなり早い時期だった）．当時の演算器のスピードは，たとえスーパ・コンピュータ用のものを用いたとしても，今日のパソコンに入っているものより遅かったのです．それを機に，いつも高速な演算回路を実現することばかり考えていたら，いつのまにか10年の歳月が流れ，ゲーム機ブームが到来しました．筆者にとってこの10年は，演算器の基礎を築くうえで貴重な準備期間でした．ちょっと先走りすぎて，今日ではあたりまえに使われている技術を当時提案し，関係者にいくら説明しても納得してもらえなかったこともあります．現在も筆者は，演算器の応用について視点を変えながら，日々考え続けています．

LSI設計を取り巻く環境も進歩し続け，変化しています．本稿および第2章で筆者がお話しする内容は非常に基本的なことですが，読者のみなさん自身が問題に直面したときに解決のヒントになると思います．

● RTL設計者には演算器とその回路の知識が必要

LSI開発の中で演算器がらみの設計を行う場合，最近ではハードウェア記述言語（HDL：hardware description language）によって，RTL（register transfer level）で記述して設計することが多くなってきています．

そして，わざわざ独自の演算器構成を考えなくても，プログラミングで演算を記述するのと同じ感覚で，ツール・ベンダなどが提供する加減乗除算程度の演算器のライブラリを利用して設計している技術者も多いと思います．どの程度最適化されているかは疑問の残るところですが，演算器の具体的な構成技術について詳しく知らなくても設計できるわけです（**図2**）．

ハードウェア化することで，ソフトウェアのときよりも楽に並列に演算を行えるようになります．ただし，演算の基本ブロックをあたりまえに思われている加減乗除に限定してしまうと，あまり最適とは言えません．

従来，コアとなる演算器の設計はASIC（application specific integrated circuit）設計に携わる技術者が行っていました．昨今のように，大規模なFPGA（field programmable gate array）が手元で扱いやすくなってくると，エンジニアのみなさんが思い思いの演算器を自由に設計できるようになってきます．そして目的の演算機能を最適に実行するような専用演算器設計を考えると，これらを自由に組み合わせた新しい演算器設計法が生まれます．そのときに，加減乗除を演算の基本ブロックと考えて構成されがちな従来の設計とは異なる発想で，最適回路を設計してみようというのが筆者の考えです．それは，小学校のときに習った筆算の演算方式とディジタル回路特有の数表現の基本に戻って，演算器群を構成するように設計を見直すということです．

参考・引用*文献

(1) Neil H. E. Weste, David Harris；CMOS VLSI Design：A Circuits and Systems Perspective（3rd Edition），Addison Wesley，May 2004.（Second Editionとはかなり内容が異なる）
(2) 大嶋洋一；エンジニアのための知的財産権概説，CQ出版社，2004年5月.

とのむら・もとのぶ
大日本印刷（株）電子デバイス事業部 電子デバイス研究所